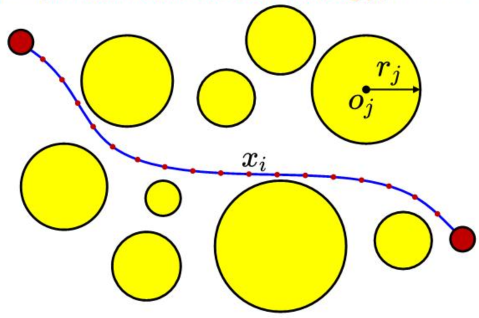
Numerical Optimization in Robotics

Homework_2 Hints

1. Formulation derivation

Cubic Spline (Minimum Stretch Energy) Path Generation



Complete the smooth path generation by unconstrained minimization:

$$\min_{x_1,x_2\ldots,x_{N-1}} \ \mathrm{Energy}(x_1,x_2\ldots,x_{N-1})+\mathrm{Potential}(x_1,x_2\ldots,x_{N-1})$$

The non-smooth path can provide an initial guess for decision variables.

The path is parameterized by N cubic spline curves. These curves are connected end to end. For the ith curve, its expression is assumed as

$$p_i(s)=a_i+b_is+c_is^2+d_is^3, s\in[0,1] \tag{1}$$

According to the assumptions of cubic spline curves with natural boundary conditions, we have

$$C^2\text{:}\ \ p_{i-1}(1)=p_i(0), p_{i-1}^{(1)}(1)=p_i^{(1)}(0), p_{i-1}^{(2)}(1)=p_i^{(2)}(0), \tag{2}$$

$$p_0^{(1)}(0)=0,\ \ p_n^{(1)}(1)=0 \tag{3}$$

1.1 Energy function

The energy function of the optimization problem is

$$Energy(x_1,x_2,\ldots,x_{n-1})=\sum_{i=0}^{n}\int_0^1\|\,p_i^{(2)}(s)\,\|^2\ ds \tag{4}$$

Let $f_i=\int_0^1\left\|p_i^2(s)\right\|^2 ds$, we then obtain the following results by some simple calculation

$$\sum_{i=0}^{n}f_i=\sum_{i=0}^{n}12d_i^2+12c_id_i+4c_i^2 \tag{5}$$

Furthermore, the derivative of f_i with respect to implicit variable x (vector) can be derived by

$$\frac{df_i}{dx} = 24\frac{dd_i}{dx}d_i + 12\frac{dc_i}{dx}d_i + 12\frac{dd_i}{dx}c_i + 8\frac{dc_i}{dx}c_i \tag{6}$$

In order to obtain the expression of $\frac{df_i}{dx}$, we have to know two quantities $\frac{dc_i}{dx}$ and $\frac{dd_i}{dx}$.

A cubic curve sequentially crossing $x_0, x_1, \ldots, x_N$ is given by

$$\begin{aligned} a_i &= x_i \\ b_i &= D_i \\ c_i &= 3(x_{i+1} - x_i) - 2D_i - D_{i+1} \\ d_i &= 2(x_i - x_{i+1}) + D_i + D_{i+1} \end{aligned}$$

where

$$\begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ \vdots \\ D_{n-2} \\ D_{n-1} \end{bmatrix} = \begin{bmatrix} 4 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 3(x_2 - x_0) \\ 3(x_3 - x_1) \\ 3(x_4 - x_2) \\ 3(x_5 - x_3) \\ \vdots \\ 3(x_{n-1} - x_{n-3}) \\ 3(x_n - x_{n-2}) \end{bmatrix}, \text{ and } D_0 = D_N = 0$$

According to the chain rule of derivation

$$\frac{dd_i}{dx} = 2\frac{d(x_i - x_{i+1})}{dx} + \frac{dD_i}{dx} + \frac{dD_{i+1}}{dx} \tag{7}$$

$$\frac{dc_i}{dx} = 3\frac{d(x_{i+1} - x_i)}{dx} - 2\frac{dD_i}{dx} - \frac{dD_{i+1}}{dx} \tag{8}$$

In which,

$$\frac{d}{dx}\begin{pmatrix} x_0 - x_1 \\ \cdots \\ x_{i-1} - x_i \\ \cdots \\ x_{n-1} - x_n \end{pmatrix}_{n\times 1} = \begin{pmatrix} 1 & -1 & & & & & \\ & 1 & -1 & & & & \\ & & 1 & -1 & & & \\ & & & 1 & -1 & & \\ & & & & \ddots & \ddots & \\ & & & & & 1 & -1 \\ & & & & & & 1 \end{pmatrix}_{n\times(n-1)} \tag{9}$$

Let's simplify the symbols as follow

$$B_{(n-1)\times 1} = 3\begin{bmatrix} x_2 - x_0 \\ x_3 - x_1 \\ x_4 - x_2 \\ x_5 - x_3 \\ \cdots \\ x_{n-1} - x_{n-3} \\ x_n - x_{n-2} \end{bmatrix}_{(n-1)\times 1} \qquad D_{(n-1)\times 1} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ \vdots \\ D_{n-2} \\ D_{n-1} \end{bmatrix}_{(n-1)\times 1} \tag{10}$$

$$A_{(n-1)\times(n-1)} = \begin{pmatrix} 4 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 4 \end{pmatrix}_{(n-1)\times(n-1)}$$

Then we can get $D = A^{-1}B$, since A is a constant matrix

$$\frac{dB}{dx} = \begin{pmatrix} 0 & 3 & & & & & \\ -3 & 0 & 3 & & & & \\ & -3 & 0 & 3 & & & \\ & & -3 & 0 & 3 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -3 & 0 & 3 \\ & & & & & -3 & 0 \end{pmatrix}_{(n-1)\times(n-1)} \tag{11}$$

So the derivative of D_i with respect to x(vector) can be represented as

$$\frac{dD}{dx} = A^{-1}\frac{dB}{dx} = A^{-1}\begin{pmatrix} 0 & 3 & & & & & \\ -3 & 0 & 3 & & & & \\ & -3 & 0 & 3 & & & \\ & & -3 & 0 & 3 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -3 & 0 & 3 \\ & & & & & -3 & 0 \end{pmatrix}_{(n-1)\times(n-1)} \tag{12}$$

Then we can get the grad of the energy function.

1.2 Potential function

$$\text{Potential}(x_1, x_2 \ldots, x_{N-1}) = 1000 \sum_{i=1}^{N-1} \sum_{j=1}^{M} \max(r_j - \|x_i - o_j\|,\ 0)$$

This function is non-zero only when the path collides with the obstacle, so if the front end uses a collision-free method such as A* this function will not work.

When the path is not collision-free,(we can use Euclidean norm)

$$P = 1000\sum_{i=1}^{N-1}\sum_{j=1}^{M} r_j - \sqrt{(x_i - a_j)^2 + (y_i - b_j)^2} \tag{13}$$

where x and y are the two dimensions of and the above equation is derived for the variables as follows

$$\frac{dP}{dx_i} = -1000\sum_{j=1}^{M}\frac{x_i - a_j}{\sqrt{(x_i - a_j)^2 + (y_i - b_j)^2}} \tag{14}$$

$$\frac{dP}{dy_i} = -1000\sum_{j=1}^{M}\frac{y_i - a_j}{\sqrt{(x_i - a_j)^2 + (y_i - b_j)^2}} \tag{15}$$

Then we can get the grad of the potential function.

2. Supplementary

- You should check your optimizer carefully before the second homework to make sure it can work correctly.
- You can try to print the value of the object function in your program to check the result.
- You have to try plenty of path to ensure your program is robust enough to handle any possible case.